Memorandum 2001 (December 2012). ISSN 1874-4850. Available from: http://www.math.utwente.nl/publications
Department of Applied Mathematics, University of Twente, Enschede, The Netherlands

An Optimal Query Assignment for Wireless Sensor Networks

Mihaela Mitici[a], Martijn Onderwater[b], Maurits de Graaf[a,c], Jan-Kees van Ommeren[a], Nico van Dijk[a], Jasper Goseling[a]

[a] *Department of Applied Mathematics, University of Twente, P.O.Box 217, 7500 AE Enschede, The Netherlands*
[b] *Centrum voor Wiskunde en Informatica, P.O.Box 94079, NL-1090 GB Amsterdam, The Netherlands*
[c] *Thales B.V., P.O.Box 88 1270, Huizen, The Netherlands*

Abstract

With the increased use of large-scale real-time embedded sensor networks, new control mechanisms are needed to avoid congestion and meet required Quality of Service (QoS) levels. In this paper, we propose a Markov Decision Problem (MDP) to prescribe an optimal query assignment strategy that achieves a trade-off between two QoS requirements: query response time and data validity. Query response time is the time that queries spend in the sensor network until they are solved. Data validity (freshness) indicates the time elapsed between data acquisition and query response and whether that time period exceeds a predefined tolerance. We assess the performance of the proposed model by means of a discrete event simulation. Compared with three other heuristics, derived from practical assignment strategies, the proposed policy performs better in terms of average assignment costs. Also in the case of real query traffic simulations, results show that the proposed policy achieves cost gains compared with the other heuristics considered. The results provide useful insight into deriving simple assignment strategies that can be easily used in practice.

Keywords: Wireless Sensor Networks, Markov Decision Processes, Quality of Service

1. Introduction

Wireless sensor networks (WSN) are intended for sensing environmental phenomena and communicating the sensed data for further use. Applications of such sensor networks include forest fire detection, intruder detection and localization and indoor environmental control [1]. The increased computing capabilities of the modern sensor networks have enabled the WSNs to become an integrated platform, where local query processing is performed. Consequently, not only the Database (DB) is able to respond to queries on the sensed environment, but also the sensors within the WSN. However, an increased number of WSN queries poses scalability and Quality of Service (QoS) challenges.

In recent years, studies in sensor networks focused on energy efficient data transmission [2, 3, 4] and the traffic was assumed to have unconstrained delivery requirements. However, growing

Email addresses: `M.A.Mitici@utwente.nl` (Mihaela Mitici), `M.Onderwater@cwi.nl` (Martijn Onderwater), `M.deGraaf@utwente.nl` (Maurits de Graaf), `J.C.W.vanOmmeren@utwente.nl` (Jan-Kees van Ommeren), `N.M.vanDijk@utwente.nl` (Nico van Dijk), `J.Goseling@utwente.nl` (Jasper Goseling)

interest in applications with specific QoS requirements has posed additional challenges. We refer to [2, 5] for an extensive outline of WSN specific QoS requirements. Literature reveals related work on QoS-based routing protocols within the sensor network. Most such protocols satisfy end-to-end packet delay [6] or data reliability requirements [7, 8] or a trade-off between the two [9]. Little work exists, however, on QoS guarantees in the field of sensor query monitoring, as addressed in this paper. In [10] a query optimizer is used to satisfy query delay requirements. In [11] the authors use data validity restrictions to specify how much time is allowed to pass since the last sensor acquisition so that the sensors are not activated, but previous sensed data is used.

This paper addresses the trade-off between two QoS requirements, namely query response time and data validity. Query response time is the time that queries spend in the WSN until they are solved. Data validity indicates the time since data stored in the DB was acquired from the WSN and whether that time exceeds a predefined tolerance. We propose a model in which response time requirements are met by ensuring timely delivery of the sensed data either directly from the sensors (WSN) or from a storage facility (DB), see Figure 1. Assigning all queries to the WSN leads to large response times. To prevent this, sensed data is reported to the DB by the sensors and queries are answered with reported data. In this case, the stored data provided to the query may exceed validity tolerances. We use Markov Decision Processes (MDP) to compute an optimal query assignment policy such that a trade-off between query response time and data validity is achieved.

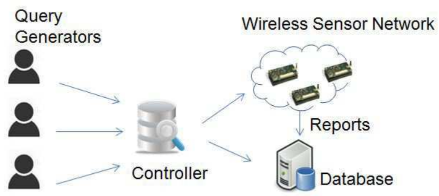


Figure 1: WSN seen as an integrated platform, where queries generated by the end-users can be solved either by the WSN or by the DB

The remainder of this paper is structured as follows. In Section 2 we describe the model of the query assignment problem and define it as a Discrete Time MDP. In Section 3 simulation results on the performance of the optimal policy in comparison to other feasible heuristic are presented. Concluding remarks and an outline of future research directions are provided in Section 4.

2. Model Formulation

2.1. Model Description

Consider a system consisting of a service facility (WSN) with processor sharing capabilities and a storage facility (DB). Figure 2 shows the proposed model. The processor sharing type of service assumed for the WSN reflects the IEE 802.15.4 MAC design principle of distributing the processing capacity fairly among the jobs simultaneously present in the network. Processor sharing service discipline for WLAN is validated by simulation in [12]. The same service discipline is assumed in [11] for query processing in WSNs.

Two types of jobs, queries and reports, arrive at the system according to a Poisson process. Queries arrive with rate λ_1. Reports arrive with rate λ_2. The service requirements of the jobs are exponentially distributed with parameter μ, independent of the job type.

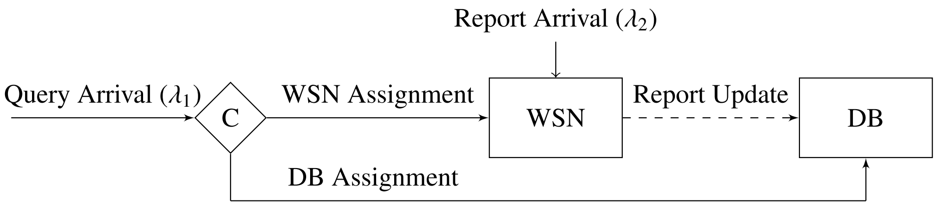


Figure 2: Proposed model with a central controller (C), the database (DB) and the wireless sensor network (WSN).

After a report is processed, the sensed data is sent to the DB for long-term storage. Incoming queries are handled by a central query controller which can assign the query either to the DB or to the WSN. When assigned to the DB, queries are immediately answered with stored data. If assigned to the WSN, queries wait to receive the sensed data, sharing the service with the other jobs present in the network. Therefore, the assignment decision is based on the trade-off between the two aforementioned QoS requirements. Our goal is to find an optimal assignment policy that achieves a trades-off between query response time and data validity.

2.2. *Stochastic Dynamic Programming Formulation*

In this section, we formulate the assignment problem as a Markov Decision Problem. In subsection 2.2.1, we consider a Continuous Time Markov Decision Process (CTMDP) with a drift. For computational reasons, we construct an exponentially uniformized Markov Decision Process in subsection 2.2.2. This leads to the formulation of the assignment problem as a Discrete Time Markov Decision Problem in subsection 2.2.3.

2.2.1. *Continuous Time Markov Decision Process with a Drift*

Consider the following Continuous Time Markov Decision Process (CTMDP) with a drift,

- State space $S = \mathbb{N}_0 \times \mathbb{N}_0 \times (0, \infty)$, where $(i, j, t) \in S$ denotes the state in which there are i queries, j reports and t the time since last report completion.
- Action: when the system is in state $(i, j, t) \in S$, the controller takes an action from the action space $D = \{DB, WSN\}$, where $d = DB$ denotes a DB assignment and $d = WSN$ denotes a WSN assignment.
- The transition rates, when in state $(i, j, N) \in S$ and action $d \in D$ is taken, are as follows,

$$q^d[(i,j,t),(i,j,t)'] = \begin{cases} \lambda_1, & (i,j,t)' = (i+1,j,t), \qquad d = WSN \\ \lambda_1, & (i,j,t)' = (i,j,t), \qquad d = DB \\ \lambda_2, & (i,j,t)' = (i,j+1,t) \\ \mu\phi_1(i,j), & (i,j,t)' = (i-1,j,t), i > 0 \\ \mu\phi_2(i,j), & (i,j,t)' = (i,j-1,0), j > 0 \end{cases}$$

where $\phi_1(i,j) = \frac{i}{i+j}$, $\phi_2(i,j) = \frac{j}{i+j}$ due to the Processor Sharing service discipline assumed for the WSN. The first and second line of the expression model a query arrival under action d. The third line models a report arrival. The last two lines model a query and report completion, respectively. In addition, there is a deterministic drift for the age component t, which linearly increases as long as no report is completed (see [13]).

Formally, the dynamics of this controlled Markovian decision process is uniquely determined by its infinitesimal generators, [14]. For the described system under action d, this generator is specified, for any arbitrary function $f : S \times S \times (0,\infty) \to \mathbb{R}$, as follows,

$$\mathbf{A}^d f(i,j,t) = \sum_{(i,j,t)'} q^d[(i,j,t),(i,j,t)'] \cdot f[(i,j,t)'] + \frac{d}{dt} f(i,j,t)$$

The generator shows that a jump to a new state $(i,j,t)'$ occurs with rate q^d or no jump occurs, but the time evolves.

- Cost: when in state (i,j,t), it is incurred a cost rate i for the queries waiting in the WSN and an instantaneous cost $(t-T)^+$, $x^+ = \max(x,0)$ upon a DB assignment.

2.2.2. Exponentially Uniformized Markov Decision Process

The assignment model is described in subsection 2.2.1 by a Markov Decision Process with a drift. Alternatively, we could use a time discretization approach as in [15]. However, both approaches would lead to technical weak convergence results and no computational results can be obtained directly as the process would have a continuous state component. For computational purposes, therefore, we artificially construct an exponentially uniformized Markov Decision Process, which will lead to both a discrete time and a discrete state MDP, as follows.

Let B an arbitrary large finite number with $B \geq \lambda_1 + \lambda_2 + \mu$. At exponential times with parameter B, the system will have a transition. Denote by s the exponential realization of a transition. Given the state space assumed in subsection 2.2.1, the transition probabilities under action $d \in D$ are as follows,

$$P^d[(i,j,t),(i,j,t)'] = \begin{cases} \lambda_1 B^{-1}, & (i,j,t)' = (i+1,j,t+s), \quad d = WSN \\ \lambda_1 B^{-1}, & (i,j,t)' = (i,j,t+s), \quad d = DB \\ \lambda_2 B^{-1}, & (i,j,t)' = (i,j+1,t+s) \\ \mu B^{-1}\phi_1(i,j), & (i,j,t)' = (i-1,j,t+s), i > 0 \\ \mu B^{-1}\phi_2(i,j), & (i,j,t)' = (i,j-1,0), j > 0 \\ 1 - (\lambda_1 + \lambda_2 + \mu \mathbf{1}_{i+j>0})B^{-1}, & (i,j,t)' = (i,j,t+s) \\ 0, & \text{otherwise} \end{cases}$$

It can be shown that the infinitesimal generators of this exponentially uniformized Markov Decision Process are the same as for the original Continuous Time Markov Decision Process (see Appendix A). Therefore, the constructed process and the original one are stochastically equivalent (see [14]).

Now observe that the time component t of the state space (i,j,t) becomes essentially a succession of exponential phases. As a Markov Process, therefore, it is sufficient to keep track of the number of exponential phases N, instead of t, the time since last report completion. The number of exponential phases approximates the time until a report completion by $t + s = (N+1) \cdot B^{-1}$.

We can now restrict ourselves to a discrete state space Markov Decision Problem, with $S = \mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{N}_0$, where $(i, j, t) \in S$ denotes the state in which there are i queries, j reports and N steps since last report completion.

2.2.3. Discrete Time Markov Decision Problem

Based on the exponentially uniformized model in Section 2.2.2, we formulate our assignment problem as a Discrete Time Markov Decision Problem (DTMDP) as follows,

- State space: $S = \mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{N}_0$, where $(i, j, N) \in S$ denotes the state with i queries and j reports in the WSN and N the age of the stored data, where N the number of steps (exponentially distributed with the uniformization parameter) since the last report completion.

- Action space: When in state (i, j, N), the query controller takes an action from the action space $D = \{DB, WSN\}$, where $d = DB$ denotes a DB assignment and $d = WSN$ denotes a WSN assignment.

- Transition probabilities, when the system is in state $(i, j, N) \in S$ and action $d \in D$ is taken, are as follows,

$$P^d[(i, j, N), (i, j, N)'] = \begin{cases} \lambda_1', & (i, j, N)' = (i+1, j, N+1), \quad d = WSN \\ \lambda_1', & (i, j, N)' = (i, j, N+1), \quad d = DB \\ \lambda_2', & (i, j, N)' = (i, j+1, N+1) \\ \mu'\phi_1(i, j), & (i, j, N)' = (i-1, j, N+1), i > 0 \\ \mu'\phi_2(i, j), & (i, j, N)' = (i, j-1, 0), j > 0 \\ 1 - (\lambda_1' + \lambda_2' + \mu'\mathbf{1}_{i+j>0}), & (i, j, N)' = (i, j, N+1) \\ 0, & \text{otherwise} \end{cases}$$

  with $\phi_1(i, j) = \frac{i}{i+j}$, $\phi_2(i, j) = \frac{j}{i+j}$ and $\lambda_i' = \lambda_i B^{-1}, i \in \{1, 2\}$ and $\mu' = \mu B^{-1}$ as per uniformization (see subsection 2.2.2). The first two lines of the expressions model query arrivals under action d. The third line models report arrivals. The forth and fifth line model query and report completions, respectively. The sixth line is a dummy transition as result of the uniformization. The last line prohibits any other state transition. Notice that every step, the age is incremented and at a report completion, the age is set to zero.

- Cost function: The system incurs a cost i for the queries waiting in the WSN. This can be interpreted as, each step (of expected length B^{-1}), the system 'pays' one unit for each waiting query. At the end of a query's service, the system would had payed one unit for each step the query was in the system, i.e. the query response time. If a query is assigned to the DB, an instantaneously penalty $C = \max(N-T)^+, (x)^+ = \max\{0, x\}$, is incurred for exceeding the validity tolerance of the stored data, where T is a predefined time tolerance. In this case, the system 'pays' one unit for each step the data validity is exceeded. Therefore, when the system is in state (i, j, N), the cost incurred per step is:

$$C(i, j, N) = i + \lambda_1'(N-T)^+\mathbf{1}_{(d=DB)}, \text{ where } (x)^+ = \max\{0, x\}.$$

Now, the quadruple (S, D, P, C) completely describes the DTMDP.
To determine an optimal assignment policy and to use standard dynamic programming, define

$$\mathbf{V}_n(i, j, N) := \text{minimal expected assignment cost over } n \text{ steps starting in state}(i, j, N).$$

Then $\mathbf{V}_n(i, j, N)$ can be computed recursively by means of the value iteration algorithm (see [16] Section 8.5.1). Consider $\mathbf{V}_0(i, j, N) = 0$ and use the following backward recursive equation,

$$\mathbf{V}_{n+1}(i, j, N) = \begin{cases} i + \lambda_1' \min \begin{cases} V_n(i+1, j, N+1) \\ (N-T)^+ + V_n(i, j, N+1) \end{cases} \\ +\lambda_2' V_n(i, j+1, N+1) \\ +\mu' \phi_1(i, j) V_n(i-1, j, N+1) \mathbf{1}_{i>0} \\ +\mu' \phi_2(i, j) V_n(i, j-1, 0) \mathbf{1}_{j>0} \\ +[1 - (\lambda_1' + \lambda_2' + \mu' \mathbf{1}_{i+j>0})] V_n(i, j, N+1). \end{cases} \tag{1}$$

The first term of the right-hand side of (1) is the cost of having i queries in service and a query assignment either to the WSN or to the DB. The next three terms represent the cost incurred by a transition due to a report arrival, a query completion and a report completion, respectively. Finally, the last term is the dummy term due to uniformization.

Simultaneously to computing $\mathbf{V}_n(i, j, N)$, the algorithm computes a ϵ-optimal stationary policy π_n which associates an optimizing action to the right-hand side of (1) for any state (i, j, N). Given the assignment policy, it is possible to compute the average assignment cost. Denote the minimal average assignment cost by $g^* = \lim_{n\to\infty}[V_{n+1}(i, j, N) - V_n(i, j, N)]$, [17]. Since the underlying Markov chain is ergodic, g^* is independent of the initial state. We approximate g^* using the following bounds introduced in [17],

$$L_n' \leq g^* \leq L_n'', \text{ where}$$

$$L_n' = \min[V_{n+1}(i, j, N) - V_n(i, j, N)], \ L_n'' = \max[V_{n+1}(i, j, N) - V_n(i, j, N)].$$

More precisely, the optimal cost g^* is computed with an accuracy ϵ by iterating the right-hand side of (1) until $L''_n - L_n' \leq \epsilon$. Then the average assignment cost is approximated as $g^* \sim (L''_n + L_n')/2$. It can be shown that the lower and upper bound converge in a finite number of steps (Theorem 8.5.4 [16]) to the optimal cost.

3. Simulation Results

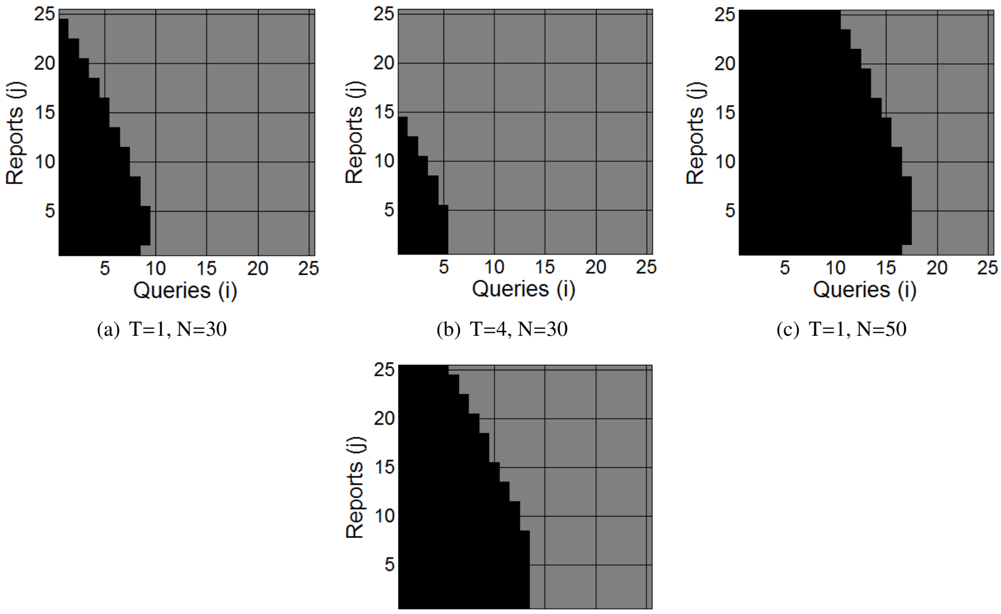
In order to illustrate the performance of the optimal policy, we present the associated average cost (g^*), as defined in Section 2, for different tolerances (T). We compare the performance of the proposed assignment policy with several fixed heuristic policies.

Figure 3 shows what action is the optimal policy associating with a state (i, j, N), when different validity tolerances are assumed.

3.1. Fixed Heuristics Policies for Performance Comparison

In order to further numerically analyze the performance of the proposed assignment strategy, we consider the following three fixed heuristic policies:

- A fixed heuristic policy π^{Db} that always assigns incoming queries to the DB. Upon a query arrival, the cost incurred is $(N - T)^+$.
- A fixed heuristic policy π^W that always assigns incoming queries to the WSN.



(a) T=1, N=30 (b) T=4, N=30 (c) T=1, N=50 (d) T=4, N=50

Figure 3: WSN assignment (black) and DB assignment (grey) assuming $\lambda_1 = 0.8$, $\lambda_2 = 0.5$ and $\mu = 1.8$.

- A heuristic policy π^T that always assigns incoming queries to the DB if the age does not exceed the tolerance, i.e. $N <= T$, and to the WSN otherwise.

Theorem 1. *Assuming the DTMDP parameters λ_1', λ_2' and μ', the average assignment cost of the heuristics π^{Db} and π^W are as follows ,*

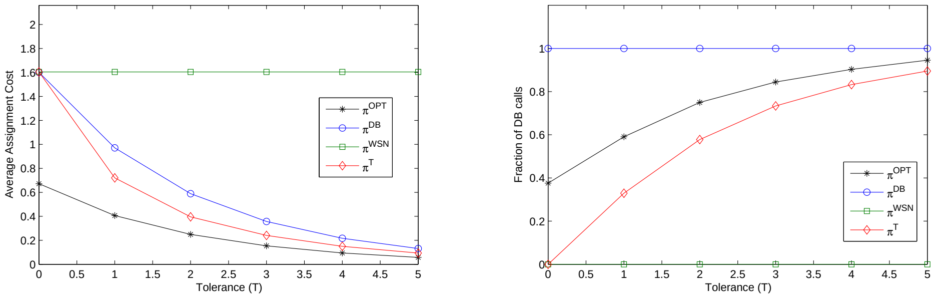
$$C_{\pi^{Db}} = \frac{\lambda_1'(1-\lambda_2')^{T+1}}{\lambda_2'} \tag{2}$$

$$C_{\pi^W} = \frac{\lambda_1'}{\mu' - (\lambda_1' + \lambda_2')} \tag{3}$$

Proof. Appendix B □

3.2. Simulation results

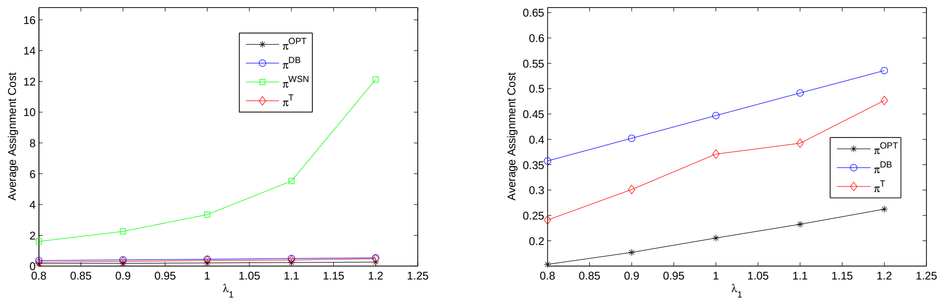
Simulation results show that, compared with the heuristics, the proposed policy achieves a lower average assignment cost (Figure 4(a)). The cost difference is significant for small time tolerances. This is of particular interest for real-time applications which specify low time tolerances. In the limit, $T \to \infty$, both π^T and π^{Db} approach the optimal policy. This can be explained by the fact that the stored data is considered valid for a longer time. Consequently, DB assignments under π^T, π^{Db} and π^{Opt} become more frequent (Figure 4(b)) since they result in small or no penalties at all. The load of the WSN is, therefore, considerably decreased.



(a) Average Assignment Cost (b) DB utilization

Figure 4: Average Assignment Cost and DB utilization assuming query and report arrival rates $\lambda_1 = 0.8$ and $\lambda_2 = 0.5$, respectively and WSN service rate $\mu = 1.8$

Also in the case of query arrival increase (Figures 5(a) and 5(b)) or processing capabilities increase, the optimal policy outperforms the heuristic policies in terms of average assignment costs (Figures 6(a) and 6(a)).

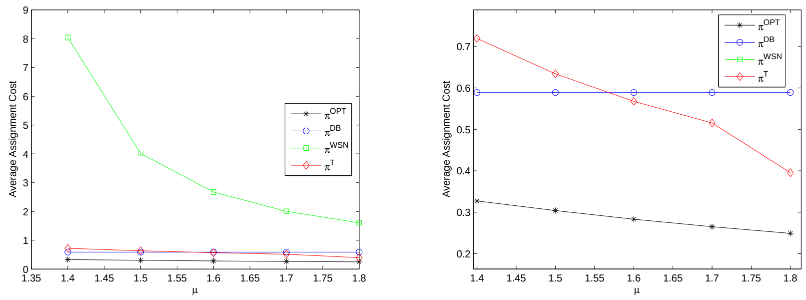


(a) Average Assignment Cost (b) Average Assignment Cost - Zoom in

Figure 5: Average Assignment Cost for different query arrival rates λ_1, $\lambda_2 = 0.5$, $\mu = 1.8$ and $T = 1$

Such insight into the performance of the system enables WSN service providers to deliver customized and efficient monitoring services to the end-users. For reasonably large data validity tolerances, simple heuristics such as π^{Db} or π^{T} perform well in comparison to the optimal policy. These are particularly suited for monitoring environments with little variation over time, e.g. temperature sensing in forests.

For applications with highly constrained delivery requirements and large data variance over time, such as fire detection or CO_2 monitoring, however, our proposed model outperforms the heuristics. Moreover, as seen in Figure 3, the optimal policy assigns incoming queries to the WSN only if the number of reports in service exceeds the number of queries. A large number of reports in service ensures frequent DB updates which, in turn, decrease the assignment costs.



(a) Average Assignment Cost

(b) Average Assignment Cost - Zoom in

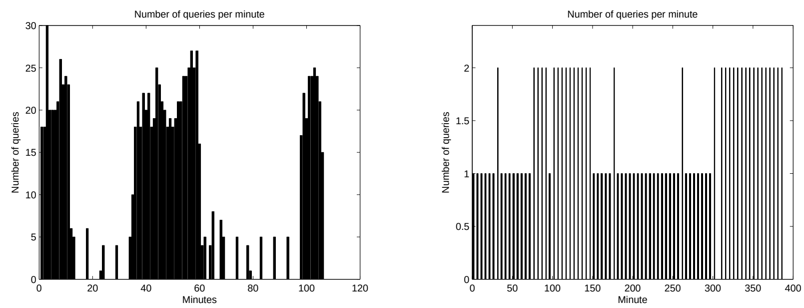
Figure 6: Average Assignment Cost for different processing capabilities μ, $\lambda_1 = 0.8$, $\lambda_2 = 0.5$ and $T = 1$

3.3. Policy Simulations for Real Query Traffic

In this subsection we assess the performance of the above described policies using data obtained from a commercial sensor network platform [18]. We use a log file containing timestamps (in seconds) of the queries arriving at the platform. We selected two time periods, depicted in Figures 7(a) and 7(b), which are representative for the intensity of query arrivals.

Dataset 1 contains timestamps of queries from one weekday around lunch time, a period when the platform typically receives many queries (see Figure 7(a)). The number of queries arriving at the platform per minute varies from about 25 in a busy period, to 5 or less in a quiet period. The coexistence of such busy periods and quiet periods makes the assumption about query arrivals originating from a homogeneous Poisson process invalid. Hence, verifying the performance of our optimal policy on such non-Poisson data provides valuable insight into its practical relevance.

Dataset 2 has timestamps of queries arriving at the platform during a night (01:00 until 07:30), see Figure 7(b). In this timespan, most queries are generated by a periodically refreshing dashboard and show little variance.

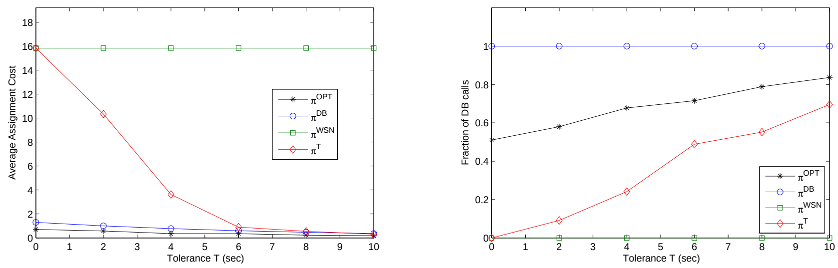


(a) Dataset 1

(b) Dataset 2

Figure 7: The number of arriving queries per minute for the two datasets.

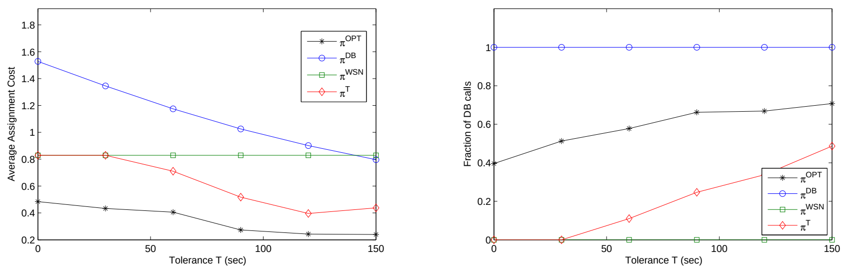
We perform a discrete event simulation and use the timestamps from the datasets as arrival moments of the queries. The optimal policy is determined using the procedure outlined in Section 2. The query arrival rate, λ_1, is estimated from the mean interarrival time of the queries in the datasets. We choose the report arrival rate λ_2 and the service rate μ such that the system has the same load as the one in Figure 4(a).



(a) Average Assignment costs (b) DB utilization

Figure 8: Average Assignment costs and DB utilization for Dataset 1

For Dataset 1, Figures 8(a) and 8(b) show that our optimal policy is better than the fixed heuristics in terms of average assignment costs. The load of the WSN is considerably decreased by routing the queries to the DB. The difference in performance is especially visible for smaller time tolerances, where the optimal policy achieves lower average costs whilst making more use of the DB. Results are similar for Dataset 2, see Figures 9(a) and 9(b). The optimal policy performs better than the fixed heuristics in terms of average assignment costs. Simulation results



(a) Average Assignment costs (b) DB utilization

Figure 9: Average Assignment costs and DB utilization for Dataset 2

show that the optimal policy achieves cost gains independent of the assumption that the arrivals at the platform follow a Poisson process. These results emphasize the applicability in practice of our proposed model. Lastly, we point out that the proposed model is independent of the WSN platform and its applications.

4. Conclusion and Future work

In this paper we investigated the trade-off between the query response time and the validity of the stored data. Firstly, we defined a Continuous Time Markov Decision Process to balance between the query response time and the validity of the stored data provided. For computational reasons, we restricted ourselves to a Discrete Time Markov Decision Problem. Given specific data validity tolerances, we provide a query assignment strategy such that the query response time is minimized. Secondly, we assessed the performance of the proposed assignment strategy in comparison to several heuristic policies by means of discrete event simulations. For low data validity tolerance, the proposed policy is shown to achieve significant cost gains in comparison to several feasible heuristics. The proposed assignment strategy outperforms the heuristics also in the case of a representative WSN platform. Future work includes enhancing the query assignment model to incorporate additional metrics such as reporting rate, reliability of the sensors, accuracy of the sensed data and energy efficiency.

Acknowledgements:

This work was performed within the project RRR (Realisation of Reliable and Secure Residential Sensor Platforms) of the Dutch program *IOP Generieke Communicatie*, number IGC1020, supported by the *Subsidieregeling Sterktes in Innovatie*.

The authors would like to thank Prof. Dr. Richard J. Boucherie, Prof. Dr. Rob van der Mei and Dr. Sandjai Bhulai for their constructive comments and suggestions.

Appendix A. Exponentially Uniformized Markov Decision Process

Uniformization is standardly known for Markov jump processes. As a drift component is introduced in the present setting, this is no longer standard.

As the infinitesimal generators uniquely define a Markov process, it is sufficient to show that the infinitesimal generators of the exponential uniformized Markov Decision Process and the original Continuous Time Markov Decision Process are identical.

Proof. To prove this, let $P^d_{\Delta t}$ denote the transition probability measures over time interval of length Δt, given that at the last jump the system is in state (i, j, t) and that upon a next jump, decision d is taken.

Let $f : \mathbb{N}\times\mathbb{N}\times\mathbb{R}$ be an arbitrary real valued function, differentiable in t and $o(\Delta t)^2 \leq Co(\Delta t)^2$ for any constant C.

Then by conditioning upon the exponential jump epoch with variable B and for arbitrary f we obtain,

$$\begin{aligned}
P^d_{\Delta t} f(i, j, t) =& e^{-\Delta t \cdot B} f(i, j, t + \Delta t) + \int_0^{\Delta t} B e^{-sB} \sum_{(i,j,t)'} P^d[(i, j, t), (i', j', t + s)] f(i', j', t + s) ds + o(\Delta t)^2 \\
&= f(i, j, t + \Delta t) - \Delta t B f(i, j, t + \Delta t) + \Delta t B \sum_{(i',j')\neq(i,j)} q^d[(i, j, t), (i', j', t)] f(i', j', t + \Delta t) B^{-1} \\
&+ \Delta t B[1 - (\lambda_1 + \lambda_2 + \mu 1_{i+j>0}) B^{-1}] f(i, j, t + \Delta t) + o(\Delta t)^2 \\
&= f(i, j, t + \Delta t) + B \sum_{(i',j')\neq(i,j)} q^d[(i, j, t), (i', j', t)][f(i', j', t + \Delta t) - f(i, j, t + \Delta t)] + o(\Delta t)^2
\end{aligned}$$

where $o(\Delta t)^2$ term reflects the probability for at least two jumps and the second term of the Taylor expansion for $e^{-\Delta B}$.

Hence, by subtracting $f(i, j, t)$, dividing by Δt and letting $\Delta t \to 0$, we obtain,

$$\begin{aligned}
\frac{P^d_{\Delta t} f(i,j,t) - f(i,j,t)}{\Delta t} =& [f(i,j,t+\Delta t) - f(i,j,t)]/\Delta t \\
&+ B[f(i,j,t+\Delta t) - f(i,j,t)] + o(\Delta t)^2 \\
&+ \sum_{(i',j')\neq(i,j)} q^d[(i,j,t),(i',j',t)][f(i',j',t) - f(i,j,t)] \\
&= \frac{d}{dt} f(i,j,t) + \sum_{(i',j')\neq(i,j)} q^d[(i,j,t),(i',j',t)][f(i',j',t) - f(i,j,t)] \\
&= A^d f(i,j,t)
\end{aligned}$$

□

Appendix B. Proof of Theorem 1

Proof. The π^W policy is independent of the validity tolerance. The WSN behaves as a regular M/M/1 PS queue. Hence, the cost of the heuristic is given by the expected number of jobs in the WSN.

$$\begin{aligned}
C_{\pi^W} &= \mathbb{E}(i) \\
&= \frac{\lambda_1}{\lambda_1 + \lambda_2} \cdot \mathbb{E}(i + j) \\
&= \frac{\lambda_1}{\lambda_1 + \lambda_2} \cdot \frac{\lambda_1 + \lambda_2}{\mu - (\lambda_1 + \lambda_2)} \\
&= \frac{\lambda_1}{\mu - (\lambda_1 + \lambda_2)}
\end{aligned} \tag{B.1}$$

We define the cost of the policy π^{Db} in terms of the limiting probabilities as follows,

$$C_{\pi^{Db}} = \lambda_1 \sum_{N \geq T} \pi_N(N) \cdot (N - T)^+, \tag{B.2}$$

where $\pi_N(N) = \sum_j \pi(j, N)$ is the long run proportion of time that the process is in state N.

We have the following balance equations for component j,

$$\begin{cases}
\pi_j(0) = \mu\pi_j(1) + (1 - \lambda_2)\pi_j(0) \\
\pi_j(1) = \mu\pi_j(2) + (1 - \lambda_2 - \mu)\pi_j(1) + \lambda_2\pi_j(0) \\
\pi_j(N-1) = \mu\pi_j(N) + (1 - \lambda_2 - \mu)\pi_j(N-1) + \lambda_2\pi_j(N-2) \\
\sum_k \pi_j(k) = 1
\end{cases} \tag{B.3}$$

where $\pi_j(0) = \sum_N \pi(0, N)$.

Solving (B.3), we have that

$$\pi_j(0) = 1 - \frac{\lambda_2}{\mu} \tag{B.4}$$

Notice that

$$\begin{aligned} \pi(0,N) &= (1-\lambda_2-\mu)\pi(0,N-1) + \mu\pi(0,N-1) \\ &= (1-\lambda_2)^N \pi(0,0) \end{aligned} \tag{B.5}$$

Now

$$\begin{aligned} \pi_j(0) &= \sum_N \pi(0,N) \\ &= \sum_N (1-\lambda_2)^N \pi(0,0) \\ &= \frac{1}{\lambda_2}\pi(0,0) \end{aligned} \tag{B.6}$$

From (B.6) and (B.4), we have that

$$\pi(0,0) = \frac{(\mu-\lambda_2)\lambda_2}{\mu} \tag{B.7}$$

We have the following balance equations for component N,

$$\begin{cases} \pi_N(0) = (1-\mu)\pi_N(N-1) + \mu\pi_N(0,N-1) \\ \pi_N(0) = \mu\sum\limits_N \pi(N) - \mu\sum\limits_N \pi(0,N), \text{ with } \pi(N) = \sum\limits_j \pi(j,N) \\ \sum\limits_k \pi_N(k) = 1 \end{cases} \tag{B.8}$$

where $\pi_N(N) = \sum\limits_j \pi(j,N)$.

But $\sum\limits_N \pi(N) = 1$ and $\sum\limits_N \pi(0,N) = \pi_j(0) = \frac{1}{\lambda_2}\pi(0,0)$ as per (B.6).

Now (B.8) become,

$$\begin{cases} \pi_N(0) = (1-\mu)\pi_N(N-1) + \mu\pi_N(0,N-1) \\ \pi_N(0) = \mu[1 - \frac{1}{\lambda_2}\pi(0,0)] \\ \sum\limits_k \pi_N(k) = 1 \end{cases} \tag{B.9}$$

Solving for (B.9), we have that

$$\pi_N(N) = \lambda_2(1-\lambda_2)^N \tag{B.10}$$

We can now compute the cost (B.2) as follows,

$$
\begin{aligned}
C_{\pi^{Db}} &= \lambda_1 \sum_{N \geq T} \pi_N(N) \cdot (N-T)^+ \\
&= \lambda_1 \sum_{N \geq T} \lambda_2 (1-\lambda_2)^N \cdot (N-T)^+ \\
&= \lambda_1 \sum_{N' \geq 0} \lambda_2 (1-\lambda_2)^{N'+T} \cdot N' \\
&= \lambda_1 \lambda_2 (1-\lambda_2)^T \sum_{N' \geq 0} (1-\lambda_2)^{N'} \cdot N' \\
&= \lambda_1 \lambda_2 (1-\lambda_2)^{T+1} \sum_{N' \geq 0} (1-\lambda_2)^{N'-1} \cdot N' \\
&= \lambda_1 \lambda_2 (1-\lambda_2)^{T+1} (-\frac{1}{\lambda_2})' \\
&= \frac{\lambda_1}{\lambda_2} (1-\lambda_2)^{T+1}
\end{aligned}
$$

$\square$

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.

[2] D. Chen and P.K. Varshney. QoS support in wireless sensor networks: A survey. In *International Conference on Wireless Networks*, pages 227–233, 2004.

[3] J. Lei, R. Yates, and L. Greenstein. A generic Model for Optimizing Single-Hop transmission Policy of Replenishable Sensors. *IEEE Transactions on Wireless Communications*, 8(2):547–551, 2009.

[4] A. Munir and A. Gordon-Ross. An MDP-based Application Oriented Optimal Policy for Wireless Sensor Networks. In *Proceedings of the 7th IEEE/ACM International Conference on Hardware/Software Codesign and System Synthesis*, pages 183–192, 2009.

[5] B. Krishnamachari. *Networking Wireless Sensors*. Cambridge University Press, 2005.

[6] T. He, J.A. Stankovic, C. Lu, and T. Abdelzaher. SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks. In *Proceedings of the 23rd IEEE International Conference on Distributed Computing Systems*, pages 46–55, 2003.

[7] B. Deb, S. Bhatnagar, and B. Nath. ReInForM: Reliable Information Forwarding Using Multiple Paths in Sensor Networks. In *Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks*, pages 406–415, 2003.

[8] F. Stann and J. Heidemann. RMST: Reliable Data Transport in Sensor Networks. In *Proceedings of the First International Workshop on Sensor Network Protocols and Applications*, pages 102–112, 2003.

[9] E. Felemban, C.G. Lee, E. Ekici, R. Boder, and S. Vural. Probabilistic QoS Guarantee in Reliability and Timeliness Domains in Wireless Sensor Networks. In *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM*, volume 4, pages 2646–2657, 2005.

[10] Y. Yao and J. Gehrke. Query Processing in Sensor Networks. In *Proceedings of the 1st Biennial Conference on Innovative Data Systems Research*, 2003.

[11] R. Khoury, T. Dawborn, B. Gafurov, G. Pink, E. Tse, Q. Tse, K. AlmiAni, M. Gaber, U. Röhm, and B. Scholz. Corona: Energy-Efficient Multi-query Processing in Wireless Sensor Networks. In *Proceedings of the Database Systems for Advanced Applications Conference*, pages 416–419, 2010.

[12] R. Litjens, F. Roijers, H. Berg, R.J. Boucherie, and M. Fleuren. Performance analysis of wireless LANs: an integrated packet/flow level approach. In *Proceedings of the 18th International Teletraffic Congress*, 2003.

[13] A. Hordijk and F.A. van der Duyn Schouten. Discretization and Weak Convergence in Markov Decision Drift Processes. *Mathematics of Operations Research*, 9(1):112–141, 1984.

[14] E.B. Dynkin. *Markov Processes*, volume 1. Academic Press, 1965.

[15] N.M. van Dijk and A. Hordijk. Time-discretization for controlled Markov processes. I. General approximation results. *Kybernetika*, 32(1):1–16, 1996.
[16] M.L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., 1994.
[17] A.R. Odoni. On finding the maximal gain for Markov Decision Processes. *Operations Research*, 17(5):857–860, 1969.
[18] Munisense. www.munisense.nl.